

# Non-Linear Post Processing Image Enhancement'

Shawn Hunt<sup>†</sup>, Alex Lopez<sup>‡</sup>, and Angel Torres<sup>‡</sup>

<sup>†</sup>University of Puerto Rico Department  
of Electrical and Computer  
Engineering  
Mayaguez, Puerto Rico, 00680  
shawn@exodo.upr.clu.edu

<sup>‡</sup>Easman Kodak Company  
1700 Dewey Avenue  
Rochester, New York 14650  
lopez@image.Kodak.COM

<sup>‡</sup>Motorola Inc.  
Land Mobile Products Sector  
8000 West Sunrise Blvd.  
Ft. Lauderdale FL, 33322  
torresan@plhp002.comm.mot.com

## Abstract

A non-linear filter for image post processing based on the feedforward Neural Network topology is presented. This study was undertaken to investigate the usefulness of "smart" filters in image post processing. The filter has shown to be useful in recovering high frequencies, such as those lost during the JPEG compression-decompression process. The filtered images have a higher signal to noise ratio, and a higher perceived image quality. Simulation studies comparing the proposed filter with the optimum mean square non-linear filter, showing examples of the high frequency recovery, and the statistical properties of the filter are given.

## 1. Introduction

An artificial neural network (NN) based image post processor for image enhancement is presented. The goal of the research was to find a practical filter which would approximate the performance of the optimum mean square non-linear filter in recovering high frequencies. This has particular relevance for compressed images, and JPEG images were chosen for some of the simulation studies.

Many lossy compression algorithms reduce or completely eliminate high frequencies during the compression-decompression process. In JPEG lossy compression for example, the spectrum of the image is quantized, and the high frequencies are usually rounded to zero. The basic problem here is then fundamentally different from the usual problem of removing noise from an image. The idea was to investigate to see whether there was information in the image that could be used to re-introduce high frequencies, and if we could implement a filter to do this. Linear filters cannot change parts of the spectrum that have been rounded to zero, and the non-linear filters available, both order statistic and others based on neural nets have, for the most part, been designed for the noise problem [1,2,3,4].

We have investigated a filter that improves the image both in terms of signal to noise ratio, and in perceived image quality. Quantization to both zero and non-zero values will cause image degradation. While linear filters may improve perceived quality caused by quantization to non-zero values, they can do nothing for values quantized to zero. Quantization to non-zero values can make the image look grainy, and linear smoothing filters can be used to improve the perceived image quality. The signal to noise ratio however, will decrease. This is acceptable, since most JPEG images are ultimately used for viewing, and perceived quality is considered to be the standard, while signal to noise ratio is used only as a secondary measure. Linear filters can only increase or decrease spectral components, and so will have no effect on frequency components rounded to zero. This is where the non-linear filters come into play. We have called these "smart" filters because they must "decide" when high frequencies have been lost, and re-introduce them. These decisions are statistically based, and must be learned during training. The first part of the research compares the optimum non-linear filter in the mean square sense to the neural filter, the neural filter is applied to a real world problem, JPEG images.

## 2. JPEG Image Compression

Without going into the details, here are the relevant aspects of the JPEG lossy standard. It might be useful to distinguish here between two variations of JPEG, the lossy and the lossless types. They are fundamentally different, and are implemented with completely different algorithms. We are interested in the lossy type here.

---

<sup>†</sup> Acknowledgments: This work was supported by NASA under Grant No. CAN NCCN-0088, the National Science Foundation under Grant No. CDA-89 13486, and the Industrial Affiliates Program at UPR.

The JPEGlossy standard proposes the following compression and decompression steps. An image is first segmented into blocks of  $8 \times 8$  pixels.  $2^{b-1}$  is subtracted from each pixel, where  $b$  is the number of bits per pixel. The dc value is calculated and compared to the dc value of the previous block. The difference between them is stored. A discrete cosine transform (DCF) is then performed on each block. The results of the DCT are then quantized using a 64 element quantization table. The values from the look up table are then coded using either arithmetic or Huffman coding. For decompression, the quantization table values are recovered from their coded values, the look up table is used to regain the quantized values, and an inverse DCT is performed.

As a result of the quantization, small values of the DCT are rounded to zero. This can manifest itself visually differently depending on the amount of quantization. An image generally has more energy in the lower frequencies than in the higher ones, which means that the higher frequency components of the DCT are usually the ones quantized to zero. A loss of sharp edges, or blurring, can be seen in the compressed-decompressed image. We are concerned with non-linear filters which can be used to remove some of the blurring effects.

### 3. Neural Filter Topology and Training

A feedforward neural net topology shown in figure 1 was chosen for the filter. Both one and two hidden layers were used in the simulations. Since they gave very similar results, we have presented only the one hidden layer case here. The number of hidden layer nodes was varied from 3 to 10.

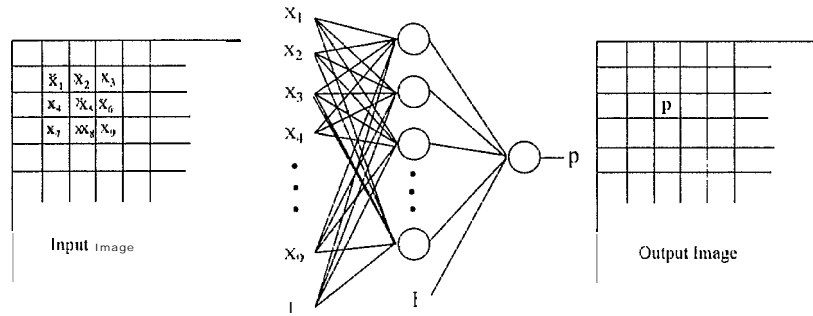


Figure 1. Topology of filter showing input and output.

The images have intensity values ranging from 0 to  $2^b - 1$ , where  $b$  is the number of bits per pixel. The images used in the simulations were 8 bits per pixel, or intensity values ranging from 0 to 255. These values were scaled in order to have inputs and outputs between 0 and 1.

Training patterns were produced using the JPEG compressed-decompressed image as input, and the original image as the target values. Let the number of rows and columns in the image be  $R$  and  $C$  respectively. Let  $p(i, j)$  be the scaled pixel in row  $i$ , column  $j$  of the original image, where  $1 \leq i \leq C$ , and  $1 \leq j \leq R$ . Also, let  $p_{JPEG}(i, j)$  denote the scaled pixel in row  $i$ , column  $j$  of the JPEG processed image. Each training pattern consists of nine input values,

$$p_{JPEG}(i, j) : k-1 \leq i \leq k+1, m-1 \leq j \leq m+1$$

and one output value,

$$p(k, m).$$

Since the inputs correspond to a pixel and its eight nearest neighbors, pixels on the borders were not used as training patterns. The method for selecting training patterns was different for the different experiments, and is described in the results section. Supervised training was done with both the Back-propagation and Levenberg-Marquardt algorithm. The Levenberg-Marquardt algorithm proved superior, and the results shown use this training method.

Once training was completed, filtering was done using a 3 by 3 pixel sliding window as shown in figure 2. All pixels except those on the border were filtered. Thus, the filtered images were 2 rows and 2 columns smaller than the original.

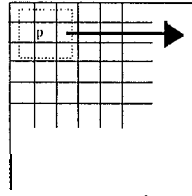


Figure 2. Filtering process using a sliding window.

## 4. Experimental Results

The non-linear filters proposed use the mean square error as a performance measure. The optimum filter would output the conditional expected value of the output pixel given the input pixels,

$$p = E\{x_5 | x_i, i=1, 2, \dots, 9\}$$

where  $p$  and  $x_i$  are as in figure 1. This can be estimated from the images to be filtered, but cannot be implemented for any but the simplest filters. For even the relatively small 3 by 3 pixel filter and 8 bit images used here, there are  $256^9$  conditional expected values. Computing and storing these is not possible on most computing platforms.

Anything over two pixels becomes computationally too expensive. The first experiment therefore uses two pixels as input. Various images were processed, using both the optimum filter and the neural filter. The high frequencies of the images were removed to produce an average signal to noise ratio of 8.2 dB relative to the original image. The complete images were used to produce the conditional expected values for the optimum filter. After filtering the average signal to noise ratio increased to 11.72 dB. The neural filter was trained using only 1/100 of the images, or 1/100 of the information used by the optimum filter. Still, the average signal to noise ratio increased to 10.1 dB after filtering. These results encouraged the use of higher order filters.

The next simulation done was to test whether a non-linear filter of this type could restore the high frequencies of a JPEG compressed image. An image was selected, and every 100<sup>th</sup> pixel was used as a training pattern. Once the training was complete, the filter was used on the same JPEG image. Figure 3 shows the results,

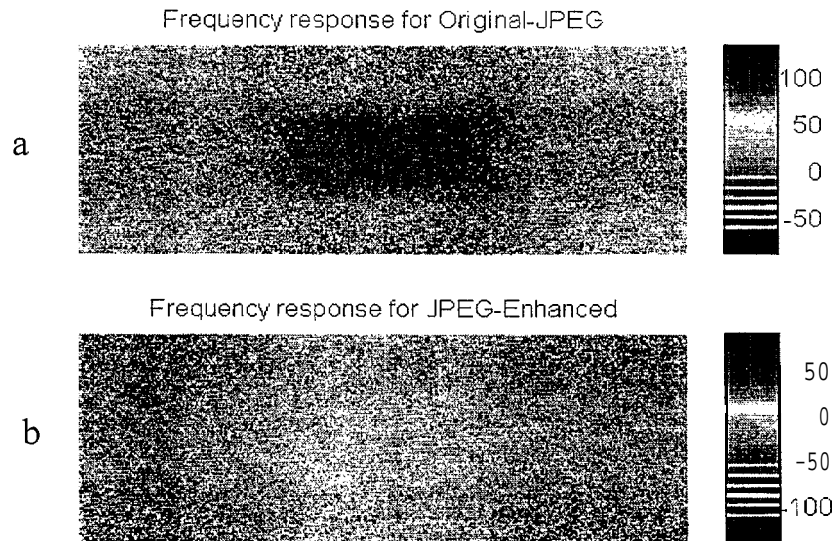


Figure 3. a) Difference of magnitude of FFT of original image and JPEG image, b) Difference of magnitude of FFT of filtered image and JPEG image

Figure 3 a is the difference between the magnitudes of the FFT of the original image, and the FFT of the JPEG compressed-decompressed image. The plot shows large differences at the high frequencies, indicating that these have been lost. The filtered image is compared to the JPEG image in figure 3 b. Again the differences are in the high frequencies, indicating that the filter did indeed introduce high frequencies. These high frequencies were not just introduced randomly, and the filtered image was better visually, and had a higher signal to noise ratio,

These results, although encouraging, are not general, since the training patterns came from the image to be filtered. A second filter was trained using training patterns from various different images. This filter was then used to filter three different images. Results of the output signal to noise ratios are shown in Table 1

Table 1. Signal to noise ratio in dB of the images before and after filtering		
	JPEG Image	Filtered Image
Image 1	29.44	30.06
Image 2	20.86	21.42
Image 3	21.94	23.18

As can be seen, the signal to noise ratio increased for all three images after filtering. More importantly, the images improved visually. In order to appreciate the difference between the images, a magnified portion of one of the images is presented in figure 4. It is clear how the JPEG compressed image is blurred compared to the original. [n figure 4 c, it can be seen that the image after filtering has edges that are more clearly defined.

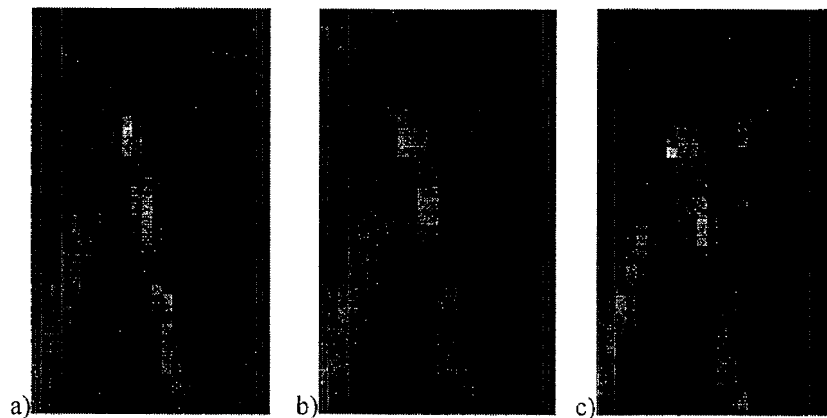


Figure 4. a) original image, b) JPEG image, c) filtered image,

The frequency plots presented above give us an idea of what the filter is doing, but the filter is not linear, so frequency response plots do not specify the filter completely. We used statistical plots in order to get a clearer picture of what the filter was accomplishing. We used two different plots for this. The first are the histograms of the original image, the image after JPEG compression, and after filtering. The plots for one of the images is shown in figure 5. The removal of the high frequencies can be seen to have the effect of smoothing the histogram.

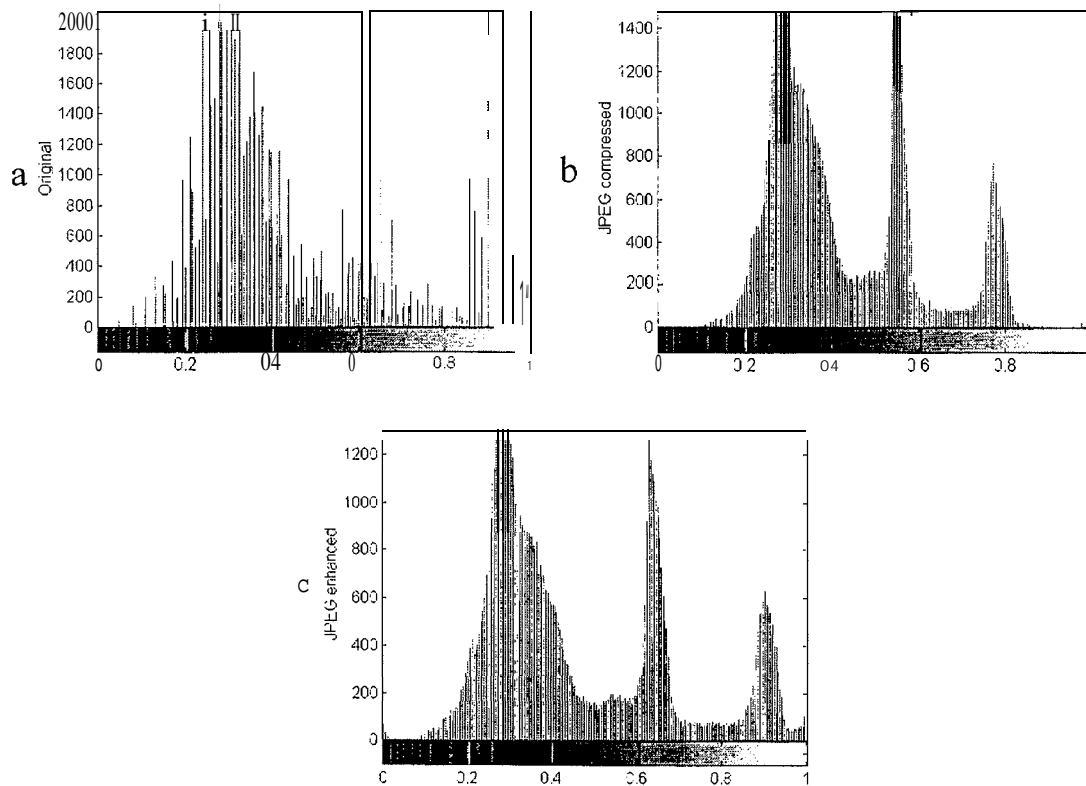


Figure 5. Histograms of a) original image, b) JPEG image c) filtered image.

The second plots were the pixel intensities before filtering vs. the pixel intensities after filtering. This will show exactly how the filter is modifying pixel intensities. If no changes had been done, then we would see only a line through the origin with slope 1. A plot for one of the images is shown in figure 6. Here it can be seen that the filter does not radically change the image, and that more mollification is done on the low pixel intensities than on the high.

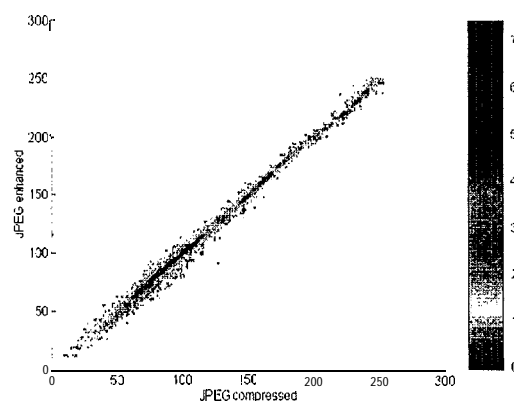


Figure 6. Plot of pixel values of the JPEG image vs. the JPEG enhanced filtered image.

## 5. Conclusions and Future Work

A non-linear filter, based on the feedforward NN topology was presented. This filter has shown to be useful in recovering high frequencies lost during the JPEG compression-decompression process. Designing filters to increase the signal to noise ratio of two dimensional signals by introducing lost high frequency components is useful in many different image processing applications. Further work needs to be done testing the filter on a wider image database, and in refining the training patterns to enhance performance.

## References

- [1] H. Hanek, N. Ansari, Z.Z. Zhang, “ Comparative study of the generalized adaptive neural filter with other non linear filters,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, p. 649-52, April 1993.
- [2] I. Pitas, “State of the art morphological and nonlinear digital image processing,” *IEEE Colloquium on ‘Morphological and Nonlinear Image Processing Techniques’*, June 1993.
- [3] A. Marston, S.-K Park, “ Nonlinear filter design using artificial neural networks,” *IEEE Proceedings of SOUTHEASTCON '91*, vol.2, pp. 931-4, April 1991.
- [4] T. Fechner, “ Nonlinear noise filtering with neural networks: comparison with Weiner optimal filtering,” *Third International Conference on Artificial Neural Networks*, p.143-7, May 1993.